

# Undecidability of the Lambek Calculus with a Relevant Modality

Max Kanovich<sup>1</sup>, Stepan Kuznetsov<sup>2</sup>, and Andre Scedrov<sup>3</sup>

<sup>1</sup>University College London;

National Research University Higher School of Economics (Moscow)

<sup>2</sup>Steklov Mathematical Institute (Moscow)

<sup>3</sup>University of Pennsylvania (Philadelphia);

National Research University Higher School of Economics (Moscow)

## Abstract

Morrill and Valentín in the paper “Computational coverage of TLG: Nonlinearity” considered an extension of the Lambek calculus enriched by a so-called “exponential” modality. This modality behaves in the “relevant” style, that is, it allows contraction and permutation, but not weakening. Morrill and Valentín stated an open problem whether this system is decidable. Here we show its undecidability. Our result remains valid if we consider the fragment where all division operations have one direction. We also show that the derivability problem in a restricted case, where the modality can be applied only to variables (primitive types), is decidable and belongs to the NP class.

## 1 The Lambek Calculus Extended by a Relevant Modality

We start with the version of the Lambek calculus,  $\mathbf{L}^*$ , that allows empty left-hand sides of sequents (introduced in [10]). We will introduce  $\mathbf{!L}^*$ —an extension of  $\mathbf{L}^*$  with one modality, denoted by  $!$ .

Formulae of  $\mathbf{!L}^*$  are built from a set of variables ( $\text{Var} = \{p, q, r, \dots\}$ ) using two binary connectives,  $/$  (*right division*) and  $\backslash$  (*left division*), and additionally one unary connective,  $!$ . Capital Latin letters denote formulae; capital Greek letters denote finite (possibly empty) *linearly ordered sequences* of formulae.

Following the linguistic tradition, formulae of the Lambek calculus (and its extensions) are also called *types*. In this terminology, variables are called *primitive types*.

---

The final publication (published in Proc. Formal Grammar 2015/2016, LNCS vol. 9804, pp. 240–256) is available at Springer via [http://dx.doi.org/10.1007/978-3-662-53042-9\\_14](http://dx.doi.org/10.1007/978-3-662-53042-9_14)

We present  $\mathbf{!L}^*$  in the form of sequent calculus. Sequents of  $\mathbf{!L}^*$  are of the form  $\Pi \rightarrow A$ , where  $A$  is a formula and  $\Pi$  is a finite (possibly empty) linearly ordered sequence of formulae.  $\Pi$  and  $A$  are called the antecedent and the succedent respectively.

The axioms and rules of  $\mathbf{!L}^*$  are as follows:

$$\begin{array}{c}
\overline{A \rightarrow A} \\
\\
\frac{\Gamma \rightarrow A \quad \Delta_1, B, \Delta_2 \rightarrow C}{\Delta_1, B / A, \Gamma, \Delta_2 \rightarrow C} (/ \rightarrow) \quad \frac{\Gamma, A \rightarrow B}{\Gamma \rightarrow B / A} (\rightarrow /) \\
\\
\frac{\Gamma \rightarrow A \quad \Delta_1, B, \Delta_2 \rightarrow C}{\Delta_1, \Gamma, A \setminus B, \Delta_2 \rightarrow C} (\setminus \rightarrow) \quad \frac{A, \Gamma \rightarrow B}{\Gamma \rightarrow A \setminus B} (\rightarrow \setminus) \\
\\
\frac{\Gamma_1, A, \Gamma_2 \rightarrow C}{\Gamma_1, !A, \Gamma_2 \rightarrow C} (! \rightarrow) \quad \frac{!A_1, \dots, !A_n \rightarrow B}{!A_1, \dots, !A_n \rightarrow !B} (\rightarrow !) \\
\\
\frac{\Delta_1, !A, \Gamma, \Delta_2 \rightarrow C}{\Delta_1, \Gamma, !A, \Delta_2 \rightarrow C} (\text{perm}_1) \quad \frac{\Delta_1, \Gamma, !A, \Delta_2 \rightarrow C}{\Delta_1, !A, \Gamma, \Delta_2 \rightarrow C} (\text{perm}_2) \\
\\
\frac{\Delta_1, !A, !A, \Delta_2 \rightarrow C}{\Delta_1, !A, \Delta_2 \rightarrow C} (\text{contr})
\end{array}$$

We call  $!$  the *relevant modality*, since it behaves in a relevant logic style, allowing contraction and permutation, but not weakening. Recall that in the original Lambek calculus there is neither contraction, nor permutation, nor weakening. The modality is introduced to restore contraction and permutation in a controlled way.

The cut rule is not officially included in  $\mathbf{!L}^*$ . Morrill and Valentín [12] claim that it is admissible and that this fact can be proved using the standard procedure (cf. [11]). In this paper we consider the system without cut and don't need its admissibility.

We also consider fragments of  $\mathbf{!L}^*$ . Since there is no cut rule in this system, it enjoys the subformula property, and therefore if we restrict the set of connectives, we obtain conservative fragments of  $\mathbf{!L}^*$ :  $\mathbf{!L}_/^*$  (where we have only  $/$  and  $!$ ),  $\mathbf{L}^*$  (this is the “pure” Lambek calculus without  $!$ ),  $\mathbf{L}_/^*$ .

As we discuss in more detail in [7],  $\mathbf{L}^*$  can be considered [1][18] as a fragment of non-commutative variant of Girard's linear logic [5]. Our modality,  $!$ , follows the spirit of the exponential connective in linear logic, allowing contraction and permutation. However, in contrast with the linear logic case, we don't allow weakening. On the other hand, as we discuss in Section 2, our  $!$  is motivated from the linguistical point of view.

**Theorem 1.** *The derivability problem for  $\mathbf{!L}^*$  is undecidable. Moreover, the derivability problem is undecidable even for  $\mathbf{!L}_/^*$ .*

*Remark 1.*  $\mathbf{!L}^*$  has been constructed as a conservative fragment of a larger system  $\mathbf{Db!}?$ , introduced in [12]. Thus Theorem 1 provides undecidability of  $\mathbf{Db!}?$  (solving an open question raised in [12]).

## 2 Linguistic Examples and Motivations

In this section we start from the standard examples of Lambek-style syntactic analysis [9][4] and then follow [12].

In syntactic formalisms based on the Lambek calculus and its variants, Lambek types (formulae) denote syntactic categories. We use the following standard primitive types:  $n$  stands for *common noun* (like “book” or “person”);  $np$  stands for *noun phrase* (like “John” or “the book”);  $s$  stands for the whole *sentence*. Actually,  $n$  and  $np$  represent not only isolated nouns and noun phrases, but also syntactic groups with similar properties: *e.g.*, “the red book” or “the person whom John met yesterday,” from the linguistic point of view, should be treated as a noun phrase ( $np$ ) as well. The latter cannot be proved to be of type  $np$  by means of  $\mathbf{L}^*$ , but  $\mathbf{!L}^*$  can handle this.

For simplicity, in our examples we don’t distinguish singular and plural forms.

Other parts of speech receive compound types:  $np \setminus s$  stands for *intransitive verb* (like “runs” or “sleeps”);  $(np \setminus s) / np$  stands for *transitive verb* (“likes,” “reads,” “met,” “admire”);  $(np \setminus s) \setminus (np \setminus s)$  is the type for *adverbs* like “yesterday” (it takes an intransitive verb group from the left-hand side and yields a compound intransitive verb group);  $np / n$  is the type for “the,” etc.

If the sequent  $A_1, \dots, A_n \rightarrow B$  is derivable in the Lambek calculus or its extension, syntactic objects of type  $A_1, \dots, A_n$ , taken together in the specified linear order, are considered to form an object of type  $B$ . For example, since  $(np \setminus s) / np, np / n, n \rightarrow np \setminus s$  is derivable, “reads the book” is an expression of type  $np \setminus s$ , or, in other words, acts as an intransitive verb.

*Example 1.*

“John met Pete.”      “John met Pete yesterday.”

These two sentences receive type  $s$ , since the sequents  $np, (np \setminus s) / np, np \rightarrow s$  and  $np, (np \setminus s) / np, np, (np \setminus s) \setminus (np \setminus s) \rightarrow s$  are both derivable in  $\mathbf{L}^*$ .

*Example 2.*

“the person whom John met”

As mentioned above, we want this phrase to receive type  $np$ . This is obtained by assigning type  $(n \setminus n) / (s / np)$  to “whom.” “John met” has type  $s / np$ , which means “a sentence that lacks a noun phrase on the right-hand side.” In other terms, we have a *gap* after “met.” In Example 1 this gap is filled by “Pete,” and here it is intentionally left blank.

*Example 3.*

“the person whom John met yesterday”

Here the gap appears in the middle of the clause (between “met” and “yesterday”), therefore “John met yesterday” is neither of type  $s/np$ , nor of type  $np \setminus s$ . This situation is called *medial extraction* and is not handled by  $\mathbf{L}^*$ .

To put  $np$  into the gap, we use  $!$  and the ( $\text{perm}_1$ ) rule:

$$\begin{array}{c}
 \frac{\frac{\frac{np, (np \setminus s) / np, np, (np \setminus s) / (np \setminus s) \rightarrow s}{np, (np \setminus s) / np, !np, (np \setminus s) / (np \setminus s) \rightarrow s} (! \rightarrow)}{\frac{np, (np \setminus s) / np, (np \setminus s) / (np \setminus s), !np \rightarrow s}{np, (np \setminus s) / np, (np \setminus s) / (np \setminus s) \rightarrow s / !np} (\text{perm}_1)} \\
 \hline
 np / n, n, n \setminus n \rightarrow np
 \end{array}$$

$$\begin{array}{ccccccc}
 np / n, & n, & (n \setminus n) / (s / !np), & np, & (np \setminus s) / np, & (np \setminus s) / (np \setminus s) & \rightarrow np \\
 \text{the} & \text{person} & \text{whom} & \text{John} & \text{met} & \text{yesterday} & 
 \end{array}$$

The sequent on top is the same schema as for “John met Pete yesterday” (see Example 1).

Note that  $s / !np$  and  $!np \setminus s$  are equivalent (due to the permutation rules).

*Example 4.*

“the paper that John signed without reading”

Finally, this is the case called *parasitic extraction*, with two  $np$  gaps (after “signed” and after “reading”). If the *that*-clause were an independent sentence, the gaps would have been filled like this: “John signed the paper without reading the paper.” To fill both gaps with the same  $np$ , we use the ( $\text{contr}$ ) rule:

$$\begin{array}{c}
 \frac{\frac{\frac{np, (np \setminus s) / np, np, ((np \setminus s) / (np \setminus s)) / np, np / np, np \rightarrow s}{np, (np \setminus s) / np, !np, ((np \setminus s) / (np \setminus s)) / np, np / np, !np \rightarrow s}}{\frac{np, (np \setminus s) / np, ((np \setminus s) / (np \setminus s)) / np, np / np, !np, !np \rightarrow s}{np, (np \setminus s) / np, ((np \setminus s) / (np \setminus s)) / np, np / np, !np \rightarrow s} (\text{contr})} \\
 \hline
 np / n, n, n \setminus n \rightarrow np
 \end{array}$$

$$\begin{array}{ccccccc}
 np / n, n, & (n \setminus n) / (s / !np), & np, & (np \setminus s) / np, & ((np \setminus s) / (np \setminus s)) / np, & np / np & \rightarrow np \\
 \text{the paper} & \text{that} & \text{John} & \text{signed} & \text{without} & \text{reading} & 
 \end{array}$$

Here “that” acts exactly as “whom,” and “without” modifies the verb group like “yesterday” does, but also requires a noun phrase “reading the paper” on the right side. The sequents on the top are easily derivable in  $\mathbf{L}^*$ .

*Remark 2.* Our calculus  $\mathbf{!L}^*$ , as well as  $\mathbf{Db!}^?$ , works well for pure complex sentences and pure compound sentences. However, we meet with difficulties in the mixed case, caused by sophisticated nature of “and” and the like. For example, the fact that “John met Pete yesterday and Mary met Ann today” has type  $s$ , leads to and unwanted classification of \* “the person whom John met yesterday and Mary met Ann today” as a noun phrase (type  $np$ ), cf. Example 3. In order to address this issue, Morrill and Valentín [12] suggest another variant of the system, denoted by  $\mathbf{Db!}^?_{\mathbf{b}}$ . This variant includes *brackets* that disallow gapping in certain situations. Morrill and Valentín pose the decidability question both for  $\mathbf{Db!}^?$  and  $\mathbf{Db!}^?_{\mathbf{b}}$ . In this paper we solve the first question.

*Remark 3.* The whole system  $\mathbf{!L}^*$  turns out to be undecidable (Theorem 1). On the other hand, notice that in these examples and the like can be treated using types of a very restricted form. Namely,  $!$  is applied only to a primitive type (for instance,  $!np$ ). In Section 5 we show that this restricted fragment of  $\mathbf{!L}^*$  is decidable. Moreover, it belongs to NP, *i.e.*, can be resolved by a nondeterministic polynomial algorithm.

### 3 $\mathbf{L}^*$ with Buszkowski's Rules

In this section we build an undecidable extension of  $\mathbf{L}^*$  with a finite set of rules, generally following the construction by W. Buszkowski from [2]. Buszkowski, however, considers another version of the Lambek calculus,  $\mathbf{L}$ , introduced in [9]. The difference between  $\mathbf{L}$  and  $\mathbf{L}^*$  is the so-called Lambek's restriction: in  $\mathbf{L}$ , the antecedents of all sequents are forced to be non-empty. In this paper, following Morrill and Valentín [12], we allow empty antecedents, and Lambek's restriction is not valid in  $\mathbf{L}^*$  (*e.g.*,  $\rightarrow p/p$  is derivable in  $\mathbf{L}^*$ ). The relationship between  $\mathbf{L}$  and  $\mathbf{L}^*$  is very subtle. For instance, the sequent  $q/(p/p) \rightarrow q$  is derivable in  $\mathbf{L}^*$ , but becomes underivable when Lambek's restriction is imposed (despite the fact that this sequent itself has a non-empty antecedent). Therefore one has to be very cautious with this issue, and for this reason here we provide a modification of Buszkowski's construction for  $\mathbf{L}^*$  rather than directly use results from [2].

Let  $\mathbf{L}^* + \mathcal{R}$  be  $\mathbf{L}^*$  extended with a finite set  $\mathcal{R}$  of rules of two special forms:

$$\frac{\Pi_1 \rightarrow p \quad \Pi_2 \rightarrow q}{\Pi_1, \Pi_2 \rightarrow r} \text{ (B}_1\text{)} \quad \text{or} \quad \frac{\Pi, q \rightarrow p}{\Pi \rightarrow r} \text{ (B}_2\text{)},$$

where  $p, q, r$  are fixed *primitive* types. We call these rules *Buszkowski's rules*.

**Theorem 2.** *The cut rule*

$$\frac{\Pi \rightarrow A \quad \Delta_1, A, \Delta_2 \rightarrow C}{\Delta_1, \Pi, \Delta_2 \rightarrow C} \text{ (cut)}$$

*is admissible in  $\mathbf{L}^* + \mathcal{R}$  for an arbitrary set  $\mathcal{R}$  of Buszkowski's rules.*

*Proof.* We proceed by double induction. We consider a number of cases, and in each of them the cut either disappears, or is replaced by cuts with simpler cut formulae ( $A$ ), or is replaced by a cut for which the depth of at least one derivation tree of a premise ( $\Pi \rightarrow A$  or  $\Delta_1, B, \Delta_2 \rightarrow C$ ) is less than for the original cut, and the cut formula remains the same. Thus by double induction (on the outer level—on the complexity of  $A$ , on the inner level—on the sum of premise derivation tree depths) we get rid of the cut.

**Case 1:**  $A$  is not the type that is introduced by the lowermost rule in the derivation of  $\Delta_1, A, \Delta_2 \rightarrow C$ . In this case (cut) can be interchanged with that lowermost rule. Consider the situation when it was (B<sub>1</sub>) (other cases are similar):

$$\frac{\Pi \rightarrow A \quad \frac{\Delta'_1, A, \Delta''_1 \rightarrow p \quad \Delta_2 \rightarrow q}{\Delta'_1, A, \Delta''_1, \Delta_2 \rightarrow r} \text{ (B}_1\text{)}}{\Delta'_1, \Pi, \Delta''_1, \Delta_2 \rightarrow r} \text{ (cut)}$$

$\}$

$$\frac{\frac{\Pi \rightarrow A \quad \Delta'_1, A, \Delta''_1 \rightarrow p}{\Delta'_1, \Pi, \Delta''_1 \rightarrow p} (\text{cut}) \quad \Delta_2 \rightarrow q}{\Delta'_1, \Pi, \Delta''_1, \Delta_2 \rightarrow r} (\text{B}_1)$$

**Case 2:**  $A = E / F$ , and it is introduced by the lowermost rules both into  $\Pi \rightarrow A$  and into  $\Delta_1, A, \Delta_2 \rightarrow C$ .

$$\frac{\frac{\Gamma, F \rightarrow E}{\Gamma \rightarrow E / F} (\rightarrow /) \quad \frac{\Pi \rightarrow F \quad \Delta_1, E, \Delta_2 \rightarrow C}{\Delta_1, E / F, \Pi, \Delta_2 \rightarrow C} (/ \rightarrow)}{\Delta_1, \Gamma, \Pi, \Delta_2 \rightarrow C} (\text{cut})$$

$\}$

$$\frac{\frac{\Pi \rightarrow F \quad \Gamma, F \rightarrow E}{\Gamma, \Pi \rightarrow E} (\text{cut}) \quad \Delta_1, E, \Delta_2 \rightarrow C}{\Delta_1, \Gamma, \Pi, \Delta_2 \rightarrow C} (\text{cut})$$

Case 2 for  $\backslash$  is handled symmetrically.

**Case 3:** one of the premises of (cut) is the axiom  $(A \rightarrow A)$ . Then the goal coincides with the other premise.

Note that since (B<sub>1</sub>) and (B<sub>2</sub>) introduce new primitive types only into the succedent, the “bad” case, where both premises of the cut rule are derived using Buszkowski’s rules and the cut formula is the formula introduced by both of them, does not occur. This is the key trick that allows to formulate the extended calculus in a cut-free way.  $\square$

In the presence of (cut) Buszkowski’s rules (B<sub>1</sub>) and (B<sub>2</sub>) are equivalent to axioms  $p, q \rightarrow r$  and  $p / q \rightarrow r$  respectively, as shown by the following derivations:

$$\frac{\Pi_1 \rightarrow p \quad \frac{\Pi_2 \rightarrow q \quad p, q \rightarrow r}{p, \Pi_2 \rightarrow r} (\text{cut})}{\Pi_1, \Pi_2 \rightarrow r} (\text{cut}) \quad \frac{\frac{\Pi, q \rightarrow p}{\Pi \rightarrow p / q} (\rightarrow /) \quad p / q \rightarrow r}{\Pi \rightarrow r} (\text{cut})$$

and in the opposite direction:

$$\frac{p \rightarrow p \quad q \rightarrow q}{p, q \rightarrow r} (\text{B}_1) \quad \frac{\frac{q \rightarrow q \quad p \rightarrow p}{p / q, q \rightarrow p} (/ \rightarrow)}{p / q \rightarrow r} (\text{B}_2)$$

From this perspective,  $\mathbf{L}^* + \mathcal{R}$  can be viewed as a finite *axiomatic extension* of  $\mathbf{L}^*$  (with non-logical axioms of a special kind). However, for our purposes it is more convenient to consider rules instead of axioms.

**Theorem 3.** *Let  $M$  be a recursively enumerable set of words over an alphabet  $\Sigma$  without the empty word. If  $\Sigma \subset \text{Var}$ , and  $\text{Var}$  also contains an infinite number*

of variables not belonging to  $\Sigma$ , then there exists a finite set  $\mathcal{R}_M$  of Buszkowski's rules and  $s \in \text{Var}$  such that for any word  $a_1 \dots a_n$  over  $\Sigma$

$$a_1 \dots a_n \in M \quad \text{iff} \quad a_1, \dots, a_n \rightarrow s \text{ is derivable in } \mathbf{L}^* + \mathcal{R}_M.$$

We shall use the fact that any recursively enumerable language without the empty word can be generated by a *binary grammar* [3]. A binary grammar is a quadruple  $G = \langle N, \Sigma, P, s \rangle$ , where  $N$  and  $\Sigma$  are disjoint alphabets ( $\Sigma$  is the original alphabet of the language),  $s \in N$ , and  $P$  is a finite set of *productions* of the form<sup>1</sup>

$$w \Rightarrow v_1 v_2 \quad \text{or} \quad v_1 v_2 \Rightarrow w,$$

where  $v_1, v_2, w \in N \cup \Sigma$ . If  $(\alpha \Rightarrow \beta) \in P$  and  $\eta$  and  $\theta$  are arbitrary (possibly empty) words over  $N \cup \Sigma$ , then  $\eta\alpha\theta \Rightarrow_G \eta\beta\theta$ . The relation  $\Rightarrow_G^*$  is the reflexive-transitive closure of  $\Rightarrow_G$ . Finally, the language generated by  $G$  is the set of all words  $a_1 \dots a_n$  over  $\Sigma$  such that  $s \Rightarrow_G^* a_1 \dots a_n$ .

*Proof.* Let  $M$  be an arbitrary recursively enumerable language without the empty word and let  $G$  be a binary grammar that generates  $M$ . We construct the corresponding extension of  $\mathbf{L}^*$ . Let  $N \cup \Sigma \subset \text{Var}$ , and let  $\text{Var}$  contain an infinite number of extra fresh variables that we'll need later. For every production  $(w \Rightarrow v_1 v_2) \in P$  we add one rule

$$\frac{\Delta_1 \rightarrow v_1 \quad \Delta_2 \rightarrow v_2}{\Delta_1, \Delta_2 \rightarrow w} \quad (\text{E})$$

For productions of the form  $v_1 v_2 \Rightarrow w$  the construction is more complex. First for every pair  $\mathbf{p} = \langle (v_1 v_2 \Rightarrow w), x \rangle$ , where  $(v_1 v_2 \Rightarrow w) \in P$  and  $x \in N \cup \Sigma$ , we introduce new variables  $\tilde{y}^{\mathbf{p}}$  for every  $y \in N \cup \Sigma$  and five extra variables  $\mathbf{a}^{\mathbf{p}}$ ,  $\mathbf{b}^{\mathbf{p}}$ ,  $\mathbf{c}^{\mathbf{p}}$ ,  $\mathbf{e}^{\mathbf{p}}$ ,  $\mathbf{f}^{\mathbf{p}}$ . Then for every  $\mathbf{p}$  we add the following rules. Some of these rules are not in Buszkowski's form. We'll transform them into the correct format below.

$$\begin{array}{ll} \frac{\Delta_1 \rightarrow \mathbf{e}^{\mathbf{p}} \quad \Delta_2 \rightarrow x}{\Delta_1, \Delta_2 \rightarrow \mathbf{a}^{\mathbf{p}}} \quad (1_{\mathbf{p}}) & \frac{\Delta_1 \rightarrow \tilde{y}^{\mathbf{p}} \quad \Delta_2, y \rightarrow \mathbf{a}^{\mathbf{p}}}{\Delta_1, \Delta_2 \rightarrow \mathbf{a}^{\mathbf{p}}} \quad (2_{\mathbf{p}}) \\ \frac{\Delta_1 \rightarrow \tilde{w}^{\mathbf{p}} \quad \Delta_2, v_1, v_2 \rightarrow \mathbf{a}^{\mathbf{p}}}{\Delta_1, \Delta_2 \rightarrow \mathbf{b}^{\mathbf{p}}} \quad (3_{\mathbf{p}}) & \frac{\Delta_1 \rightarrow \tilde{y}^{\mathbf{p}} \quad \Delta_2, y \rightarrow \mathbf{b}^{\mathbf{p}}}{\Delta_1, \Delta_2 \rightarrow \mathbf{b}^{\mathbf{p}}} \quad (4_{\mathbf{p}}) \\ \frac{\Delta_1 \rightarrow \mathbf{f}^{\mathbf{p}} \quad \Delta_2, \mathbf{e}^{\mathbf{p}} \rightarrow \mathbf{b}^{\mathbf{p}}}{\Delta_1, \Delta_2 \rightarrow \mathbf{c}^{\mathbf{p}}} \quad (5_{\mathbf{p}}) & \frac{\Delta_1 \rightarrow y \quad \Delta_2, \tilde{y}^{\mathbf{p}} \rightarrow \mathbf{c}^{\mathbf{p}}}{\Delta_1, \Delta_2 \rightarrow \mathbf{c}^{\mathbf{p}}} \quad (6_{\mathbf{p}}) \\ \frac{\Delta, \mathbf{f}^{\mathbf{p}} \rightarrow \mathbf{c}^{\mathbf{p}}}{\Delta \rightarrow x} \quad (7_{\mathbf{p}}) & \end{array}$$

---

<sup>1</sup>In the definition from [3],  $P$  could also include productions of the form  $u \Rightarrow v$  for  $u, v \in N \cup \Sigma$ . Such a rule can be equivalently replaced by two productions  $u \Rightarrow w_1 w_2$ ,  $w_1 w_2 \Rightarrow v$ , where  $w_1$  and  $w_2$  are new elements added to  $N$  (different for different rules). We encode these simple productions using more complex ones in order to reduce the number of cases to be considered in the proofs.

As already said, some of these rules are not actually Buszkowski's rules. However, any rule of the form

$$\frac{\Delta_1 \rightarrow p \quad \Delta_2, q \rightarrow r}{\Delta_1, \Delta_2 \rightarrow t}$$

(these are rules  $(2_p)$ ,  $(4_p)$ ,  $(5_p)$ , and  $(6_p)$ ) can be equivalently replaced by two rules

$$\frac{\Delta, q \rightarrow r}{\Delta \rightarrow u} (B_2) \quad \text{and} \quad \frac{\Delta_1 \rightarrow p \quad \Delta_2 \rightarrow u}{\Delta_1, \Delta_2 \rightarrow t} (B_1)$$

where  $u$  is a fresh variable.

Similarly,  $(3_p)$  is a shortcut for three rules:

$$\frac{\Delta, v_2 \rightarrow \mathbf{a}^p}{\Delta \rightarrow u_1} (B_2) \quad \frac{\Delta, v_1 \rightarrow u_1}{\Delta \rightarrow u_2} (B_2) \quad \frac{\Delta_1 \rightarrow \tilde{w}^p \quad \Delta_2 \rightarrow u_2}{\Delta_1, \Delta_2 \rightarrow \mathbf{b}^p} (B_1)$$

Rules  $(1_p)$ ,  $(7_p)$ , and (E) are already in the correct format. Thus we've actually constructed a calculus of the form  $\mathbf{L}^* + \mathcal{R}$ . Denote it by  $\mathbf{L}^* + \mathcal{R}_M$ .

Now to achieve our goal it is sufficient to prove that for any  $x, z_1, \dots, z_m \in N \cup \Sigma$

$$x \Rightarrow_G^* z_1 \dots z_m \quad \text{iff} \quad z_1, \dots, z_m \rightarrow x \text{ is derivable in } \mathbf{L}^* + \mathcal{R}_M.$$

The proof consists of two directions.

$\boxed{\Leftarrow}$  All types in the sequent  $z_1, \dots, z_m \rightarrow x$  are primitive, therefore its derivation includes only axioms and Buszkowski's rules, but not original rules of  $\mathbf{L}^*$  ( $(\rightarrow /)$ ,  $(\rightarrow \backslash)$ ,  $(/ \rightarrow)$ ,  $(\backslash \rightarrow)$ ).

Since  $\mathbf{e}^p$ ,  $\mathbf{f}^p$ , and  $\tilde{y}^p$  (for all  $y \in N \cup \Sigma$ , including  $w$ ) do not appear in the succedents of goal sequents in Buszkowski's rules from  $\mathcal{R}_M$ , the only possible situation when  $\mathbf{e}^p$ ,  $\mathbf{f}^p$ , or  $\tilde{y}^p$  actually appears in the succedent is the axiom. Hence rules  $(1_p)$ – $(5_p)$  can be rewritten in a simpler way (rules  $(6_p)$  and  $(7_p)$  are not affected by this simplification):

$$\begin{array}{lll} \frac{\Phi \rightarrow x}{\mathbf{e}^p, \Phi \rightarrow \mathbf{a}^p} (1'_p) & \frac{\Phi, y \rightarrow \mathbf{a}^p}{\tilde{y}^p, \Phi \rightarrow \mathbf{a}^p} (2'_p) & \frac{\Phi, v_1, v_2 \rightarrow \mathbf{a}^p}{\tilde{w}^p, \Phi \rightarrow \mathbf{b}^p} (3'_p) \\ \frac{\Phi, y \rightarrow \mathbf{b}^p}{\tilde{y}^p, \Phi \rightarrow \mathbf{b}^p} (4'_p) & \frac{\Phi, \mathbf{e}^p \rightarrow \mathbf{b}^p}{\mathbf{f}^p, \Phi \rightarrow \mathbf{c}^p} (5'_p) & \frac{\Delta_1 \rightarrow y \quad \Delta_2, \tilde{y}^p \rightarrow \mathbf{c}^p}{\Delta_1, \Delta_2 \rightarrow \mathbf{c}^p} (6_p) \\ \frac{\Phi, \mathbf{f}^p \rightarrow \mathbf{c}^p}{\Phi \rightarrow x} (7_p) & & \end{array}$$

Proceed by induction on the cut-free derivation. The sequent  $z_1, \dots, z_m \rightarrow x$  could either be an axiom (and then  $n = 1$ ,  $z_1 = x$ , and trivially  $x \Rightarrow_G^* x$ ) or be derived by one of the Buszkowski's rules. Since  $x \in N \cup \Sigma$ , the only possible rules are (E) and  $(7_p)$ .

If  $z_1, \dots, z_m \rightarrow x$  is derived using (E):

$$\frac{z_1, \dots, z_k \rightarrow v_1 \quad z_{k+1}, \dots, z_m \rightarrow v_2}{z_1, \dots, z_k, z_{k+1}, \dots, z_m \rightarrow x} (E),$$



then we have  $z_1, \dots, z_k \rightarrow v_1$ ,  $z_{k+1}, \dots, z_m \rightarrow v_2$ , and  $(x \Rightarrow v_1 v_2) \in P$ . By induction hypothesis,  $v_1 \Rightarrow_G^* z_1 \dots z_k$  and  $v_2 \Rightarrow_G^* z_{k+1} \dots z_m$ , therefore we get  $x \Rightarrow_G v_1 v_2 \Rightarrow_G^* z_1 \dots z_k z_{k+1} \dots z_m$ .

If the last rule in the derivation is  $(7_p)$ , then we get  $z_1, \dots, z_m, \mathbf{f}^p \rightarrow \mathbf{c}^p$ . Trace the type in the succedent. Since the antecedent doesn't contain  $\mathbf{c}^p$ ,  $\mathbf{b}^p$ , or  $\mathbf{a}^p$ , sequents with these types in the succedent could not appear as axioms, and the only ways to derive such sequents are represented by the following schema (the arrows go from goal to premises):

$$\begin{array}{c} \xrightarrow{\quad} \mathbf{c}^p \xrightarrow{(5'_p)} \mathbf{b}^p \xrightarrow{(3'_p)} \mathbf{a}^p \xrightarrow{(1'_p)} x \\ \begin{array}{ccc} \textcircled{(6_p)} & \textcircled{(4'_p)} & \textcircled{(2'_p)} \\ \curvearrowright & \curvearrowright & \curvearrowright \end{array} \end{array}$$

Therefore, the sequent  $z_1, \dots, z_m, \mathbf{f}^p \rightarrow \mathbf{c}^p$  is derived in the following way: several (possibly zero) applications of  $(6_p)$ , then  $(5'_p)$ , then several  $(4'_p)$ , then several  $(2'_p)$ , then  $(1'_p)$ . Finally, on top of this last  $(1'_p)$  rule we again get a sequent with  $x$  in the succedent. The whole derivation has the following form. Here  $*$  means several consecutive applications of the same rule, and  $\Delta_1, \dots, \Delta_n = z_1, \dots, z_m$ .

$$\begin{array}{c} \frac{\frac{\frac{\frac{\frac{y_1, \dots, y_{k-1}, v_1, v_2, y_{k+1}, \dots, y_n \rightarrow x}{\mathbf{e}^p, y_1, \dots, y_{k-1}, v_1, v_2, y_{k+1}, \dots, y_n \rightarrow \mathbf{a}^p} (1'_p)}{\widehat{y}_{k+1}^p, \dots, \widehat{y}_n^p, \mathbf{e}^p, y_1, \dots, y_{k-1}, v_1, v_2 \rightarrow \mathbf{a}^p} (2'_p)^*}{\widehat{w}^p, \widehat{y}_{k+1}^p, \dots, \widehat{y}_n^p, \mathbf{e}^p, y_1, \dots, y_{k-1} \rightarrow \mathbf{b}^p} (3'_p)}{\widehat{y}_1^p, \dots, \widehat{y}_{k-1}^p, \widehat{w}^p, \widehat{y}_{k+1}^p, \dots, \widehat{y}_n^p, \mathbf{e}^p \rightarrow \mathbf{b}^p} (4'_p)^*}{\mathbf{f}^p, \widehat{y}_1^p, \dots, \widehat{y}_{k-1}^p, \widehat{w}^p, \widehat{y}_{k+1}^p, \dots, \widehat{y}_n^p \rightarrow \mathbf{c}^p} (5'_p)}{\Delta_1 \rightarrow y_1 \quad \dots \quad \Delta_k \rightarrow w \quad \dots \quad \Delta_n \rightarrow y_n \quad \mathbf{f}^p, \widehat{y}_1^p, \dots, \widehat{y}_{k-1}^p, \widehat{w}^p, \widehat{y}_{k+1}^p, \dots, \widehat{y}_n^p \rightarrow \mathbf{c}^p} (6_p)^*} \\ \frac{\Delta_1, \dots, \Delta_n, \mathbf{f}^p \rightarrow \mathbf{c}^p}{\Delta_1, \dots, \Delta_n \rightarrow x} (7_p) \end{array}$$

Here rule  $(1'_p)$  introduces  $\mathbf{e}^p$ ,  $(2'_p)$  moves  $\mathbf{e}^p$  to the left and marks  $y_i$  as  $\widehat{y}_i^p$ ,  $(3'_p)$  actually applies the production  $(v_1 v_2 \Rightarrow w)$ , which is possible, since  $v_1, v_2$  is now on the edge of the antecedent,  $(4'_p)$  continues the movement, and finally  $(5'_p)$ ,  $(6_p)$ , and  $(7_p)$  move the letters backwards, unmark them and return the antecedent to  $x$ .

By induction hypothesis,  $y_1 \Rightarrow_G^* \Delta_1$ ,  $\dots$ ,  $y_{k-1} \Rightarrow_G^* \Delta_{k-1}$ ,  $w \Rightarrow_G^* \Delta_k$ ,  $y_{k+1} \Rightarrow_G^* \Delta_{k+1}$ ,  $\dots$ ,  $y_n \Rightarrow_G^* \Delta_n$ , and  $x \Rightarrow_G^* y_1 \dots y_{k-1} v_1 v_2 y_{k+1} \dots y_n$ . By application of  $v_1 v_2 \Rightarrow w$  we get  $x \Rightarrow_G^* \Delta_1 \dots \Delta_n$ .

We notice that the first type of productions of the binary grammar is handled much easier than the second one. This is due to the fact that in the first case we simulate standard context-free derivation, while in the second case the production is not context-free and even not context-sensitive.

$\boxed{\Rightarrow}$  Proceed by induction on  $\Rightarrow_G^*$ . For the base case  $(x \Rightarrow_G^* x)$  the corresponding sequent  $(x \rightarrow x)$  is an axiom.

If the last production is  $w \Rightarrow v_1 v_2$ :

$$x \Rightarrow_G^* z_1 \dots z_{k-1} w z_{k+1} \dots z_m \Rightarrow_G z_1 \dots z_{k-1} v_1 v_2 z_{k+1} \dots z_m,$$

then by induction hypothesis  $z_1, \dots, z_{k-1}, w, z_{k+1}, \dots, z_m \rightarrow x$  is derivable in  $\mathbf{L}^* + \mathcal{R}_M$ . Also  $v_1, v_2 \rightarrow w$  is derivable by (B<sub>1</sub>), and by (cut) we obtain  $z_1, \dots, z_{k-1},$

$v_1, v_2, z_{k+1}, \dots, z_m \rightarrow x$ . The cut rule is admissible in  $\mathbf{L}^* + \mathcal{R}$  by Theorem 2.

For the  $v_1 v_2 \Rightarrow w$  case, *i.e.*, the last production is applied like this:

$$x \Rightarrow_G^* z_1 \dots z_{k-1} v_1 v_2 z_{k+1} \dots z_m \Rightarrow_G z_1 \dots z_{k-1} w z_{k+1} \dots z_m,$$

the derivation is as follows (here  $\mathbf{p} = \langle (v_1 v_2 \Rightarrow w), x \rangle$ ):

$$\frac{\frac{\frac{\frac{z_1, \dots, z_{k-1}, v_1, v_2, z_{k+1}, \dots, z_m \rightarrow x}{\mathbf{e}^{\mathbf{p}}, z_1, \dots, z_{k-1}, v_1, v_2, z_{k+1}, \dots, z_m \rightarrow \mathbf{a}^{\mathbf{p}}} (1'_{\mathbf{p}})}{\frac{\mathbf{e}^{\mathbf{p}}, z_1, \dots, z_{k-1}, v_1, v_2, z_{k+1}, \dots, z_m \rightarrow \mathbf{a}^{\mathbf{p}}}{\mathbf{z}_{k+1}^{\mathbf{p}}, \dots, \mathbf{z}_m^{\mathbf{p}}, \mathbf{e}^{\mathbf{p}}, z_1, \dots, z_{k-1}, v_1, v_2 \rightarrow \mathbf{a}^{\mathbf{p}}} (2'_{\mathbf{p}})^*}{\frac{\mathbf{z}_{k+1}^{\mathbf{p}}, \dots, \mathbf{z}_m^{\mathbf{p}}, \mathbf{e}^{\mathbf{p}}, z_1, \dots, z_{k-1}, v_1, v_2 \rightarrow \mathbf{a}^{\mathbf{p}}}{\mathbf{w}^{\mathbf{p}}, \mathbf{z}_{k+1}^{\mathbf{p}}, \dots, \mathbf{z}_m^{\mathbf{p}}, \mathbf{e}^{\mathbf{p}}, z_1, \dots, z_{k-1} \rightarrow \mathbf{b}^{\mathbf{p}}} (3'_{\mathbf{p}})}{\frac{\mathbf{w}^{\mathbf{p}}, \mathbf{z}_{k+1}^{\mathbf{p}}, \dots, \mathbf{z}_m^{\mathbf{p}}, \mathbf{e}^{\mathbf{p}}, z_1, \dots, z_{k-1} \rightarrow \mathbf{b}^{\mathbf{p}}}{\mathbf{z}_1^{\mathbf{p}}, \dots, \mathbf{z}_{k-1}^{\mathbf{p}}, \mathbf{w}^{\mathbf{p}}, \mathbf{z}_{k+1}^{\mathbf{p}}, \dots, \mathbf{z}_m^{\mathbf{p}}, \mathbf{e}^{\mathbf{p}} \rightarrow \mathbf{b}^{\mathbf{p}}} (4'_{\mathbf{p}})^*}{\frac{\mathbf{z}_1^{\mathbf{p}}, \dots, \mathbf{z}_{k-1}^{\mathbf{p}}, \mathbf{w}^{\mathbf{p}}, \mathbf{z}_{k+1}^{\mathbf{p}}, \dots, \mathbf{z}_m^{\mathbf{p}}, \mathbf{e}^{\mathbf{p}} \rightarrow \mathbf{b}^{\mathbf{p}}}{\mathbf{f}^{\mathbf{p}}, \mathbf{z}_1^{\mathbf{p}}, \dots, \mathbf{z}_{k-1}^{\mathbf{p}}, \mathbf{w}^{\mathbf{p}}, \mathbf{z}_{k+1}^{\mathbf{p}}, \dots, \mathbf{z}_m^{\mathbf{p}} \rightarrow \mathbf{c}^{\mathbf{p}}} (5'_{\mathbf{p}})}{\frac{\mathbf{f}^{\mathbf{p}}, \mathbf{z}_1^{\mathbf{p}}, \dots, \mathbf{z}_{k-1}^{\mathbf{p}}, \mathbf{w}^{\mathbf{p}}, \mathbf{z}_{k+1}^{\mathbf{p}}, \dots, \mathbf{z}_m^{\mathbf{p}} \rightarrow \mathbf{c}^{\mathbf{p}}}{\frac{z_1, \dots, z_{k-1}, w, z_{k+1}, \dots, z_m, \mathbf{f}^{\mathbf{p}} \rightarrow \mathbf{c}^{\mathbf{p}}}{z_1, \dots, z_{k-1}, w, z_{k+1}, \dots, z_m \rightarrow x} (7_{\mathbf{p}})} (6_{\mathbf{p}})^*$$

The sequent on the top is derivable by inductive hypothesis.  $\square$

Since there exist undecidable recursively enumerable languages, Theorem 3 now yields the following result:

**Theorem 4.** *There exists a finite set of Buszkowski's rules  $\mathcal{R}_0$  such that the derivability problem for  $\mathbf{L}^* + \mathcal{R}_0$  is undecidable.*

Note that the  $\setminus$  connective is not used in the construction, so we've actually obtained undecidability for  $\mathbf{L}^*_\setminus + \mathcal{R}_0$ .

## 4 Undecidability of $\mathbf{!L}^*$

We prove undecidability of  $\mathbf{!L}^*$  by encoding  $\mathbf{L}^* + \mathcal{R}$  derivations in this calculus. In order to do that, we first prove a technical proposition.

If  $\mathcal{R}$  is a set of Buszkowski's rules, let

$$\mathcal{G}_{\mathcal{R}} = \left\{ (r/q)/p \mid \frac{\Pi_1 \rightarrow p \quad \Pi_2 \rightarrow q}{\Pi_1, \Pi_2 \rightarrow r} \in \mathcal{R} \right\} \cup \left\{ r/(p/q) \mid \frac{\Pi, q \rightarrow p}{\Pi \rightarrow r} \in \mathcal{R} \right\}.$$

If  $\mathcal{B} = \{B_1, \dots, B_n\}$  is a finite set of formulae, let  $\mathbf{!}\Gamma_{\mathcal{B}} = \mathbf{!}B_1, \dots, \mathbf{!}B_n$ . (The order of the elements in  $\mathcal{B}$  doesn't matter, since  $\mathbf{!}\Gamma_{\mathcal{B}}$  will appear in left-hand sides of  $\mathbf{!L}^*$  sequents, and in  $\mathbf{!L}^*$  we have the (perm<sub>1,2</sub>) rules.)

**Theorem 5.**  $\mathbf{L}^* + \mathcal{R} \vdash \Pi \rightarrow A$  if and only if there exists  $\mathcal{B} \subseteq \mathcal{G}_{\mathcal{R}}$  such that  $!\mathbf{L}^* \vdash !\Gamma_{\mathcal{B}}, \Pi \rightarrow A$ .

In this theorem a finite *theory* ( $\mathcal{R}$ ) that extends the basic calculus ( $\mathbf{L}^*$ ) gets embedded into the formula (more precisely, the sequent  $\Pi \rightarrow A$ ) being derived. In linear logic this is possible with the help of the exponential modality (!). However, our version of ! doesn't enjoy the weakening rule, therefore we cannot always take  $\mathcal{B} = \mathcal{G}_{\mathcal{R}}$ , as one usually could expect. Generally, with  $\mathcal{B} = \mathcal{G}_{\mathcal{R}}$  the “only if” statement is false. For example,  $!(r/(p/q)), s \rightarrow s$  is not derivable in  $!\mathbf{L}^*$ , but  $s \rightarrow s$  is indeed derivable in  $\mathbf{L}^* + \mathcal{R}$  for any  $\mathcal{R}$ . This happens because this particular Buszkowski's rule, encoded by  $r/(p/q)$ , is not *relevant* to  $s \rightarrow s$ .

*Proof.*  $\boxed{\Rightarrow}$  Proceed by induction.

If  $\Pi \rightarrow A$  is an axiom ( $A \rightarrow A$ ), just take  $\mathcal{B} = \emptyset$ .

If  $A = B/C$ , and  $\Pi \rightarrow A$  is obtained using the  $(\rightarrow /)$  rule from  $\Pi, C \rightarrow B$ , then take the same  $\mathcal{B}$  and apply the same rule:

$$\frac{!\Gamma_{\mathcal{B}}, \Pi, C \rightarrow B}{!\Gamma_{\mathcal{B}}, \Pi \rightarrow B/C}$$

If  $\Pi = \Phi_1, B/C, \Psi, \Phi_2$ , and  $\Pi \rightarrow A$  is obtained by  $(/ \rightarrow)$  from  $\Psi \rightarrow C$  and  $\Phi_1, B, \Phi_2 \rightarrow A$ , then by induction hypothesis  $!\mathbf{L}^* \vdash !\Gamma_{\mathcal{B}_1}, \Psi \rightarrow C$  and  $!\mathbf{L}^* \vdash !\Gamma_{\mathcal{B}_2}, \Phi_1, B, \Phi_2 \rightarrow A$  for some  $\mathcal{B}_1, \mathcal{B}_2 \subseteq \mathcal{G}_{\mathcal{A}}$ . Let  $\mathcal{B} = \mathcal{B}_1 \cup \mathcal{B}_2$ . Then for  $!\Gamma_{\mathcal{B}}, \Pi \rightarrow A$  we have the following derivation in  $!\mathbf{L}^*$ , where  $*$  means several applications of the rules in any order.

$$\frac{\frac{!\Gamma_{\mathcal{B}_1}, \Psi \rightarrow C \quad !\Gamma_{\mathcal{B}_2}, \Phi_1, B, \Phi_2 \rightarrow A}{!\Gamma_{\mathcal{B}_2}, \Phi_1, B/C, !\Gamma_{\mathcal{B}_1}, \Psi, \Phi_2 \rightarrow A} (/ \rightarrow)}{!\Gamma_{\mathcal{B}_1 \cup \mathcal{B}_2}, \Phi_1, B/C, \Psi, \Phi_2 \rightarrow A} (\text{contr, perm})^*$$

Finally,  $\Pi \rightarrow A$  can be obtained by application of Buszkowski's rules ( $B_1$ ) or ( $B_2$ ). In the first case,  $A = r$ ,  $\Pi = \Pi_1, \Pi_2$ , and both  $\Pi_1 \rightarrow p$  and  $\Pi_2 \rightarrow q$  are derivable in  $\mathbf{L}^* + \mathcal{R}$ . Thus by induction hypothesis we get  $!\mathbf{L}^* \vdash !\Gamma_{\mathcal{B}_1}, \Pi_1 \rightarrow p$  and  $!\mathbf{L}^* \vdash !\Gamma_{\mathcal{B}_2}, \Pi_2 \rightarrow q$  for some  $\mathcal{B}_1, \mathcal{B}_2 \subseteq \mathcal{G}_{\mathcal{R}}$ . Moreover,  $(r/q)/p \in \mathcal{G}_{\mathcal{R}}$ . Now take  $\mathcal{B} = \mathcal{B}_1 \cup \mathcal{B}_2 \cup \{(r/q)/p\}$  and enjoy the following derivation for  $!\Gamma_{\mathcal{B}}, \Pi_1, \Pi_2 \rightarrow r$  in  $!\mathbf{L}^*$ :

$$\frac{\frac{\frac{!\Gamma_{\mathcal{B}_2}, \Pi_2 \rightarrow q \quad r \rightarrow r}{r/q, !\Gamma_{\mathcal{B}_2}, \Pi_2 \rightarrow r} (/ \rightarrow)}{!(r/q)/p, !\Gamma_{\mathcal{B}_1}, \Pi_1, !\Gamma_{\mathcal{B}_2}, \Pi_2 \rightarrow r} (/ \rightarrow)}{!((r/q)/p), !\Gamma_{\mathcal{B}_1}, \Pi_1, !\Gamma_{\mathcal{B}_2}, \Pi_2 \rightarrow r} (! \rightarrow)}{!\Gamma_{\mathcal{B}}, \Pi_1, \Pi_2 \rightarrow r} (\text{contr, perm})^*$$

In the ( $B_2$ ) case,  $A = r$ , and we have  $!\Gamma_{\mathcal{B}'}, \Pi, q \rightarrow p$  in the induction hypothesis for some  $\mathcal{B}' \subseteq \mathcal{G}_{\mathcal{R}}$ . Let  $\mathcal{B} = \mathcal{B}' \cup \{r/(p/q)\}$  (recall that  $r/(p/q) \in \mathcal{G}_{\mathcal{R}}$ ),

and proceed like this:

$$\frac{\frac{\frac{! \Gamma_{\mathcal{B}'}, \Pi, q \rightarrow p}{! \Gamma_{\mathcal{B}'}, \Pi \rightarrow p / q} (\rightarrow /)}{r / (p / q), ! \Gamma_{\mathcal{B}'}, \Pi \rightarrow r} (\rightarrow /)}{\frac{!(r / (p / q)), ! \Gamma_{\mathcal{B}'}, \Pi \rightarrow r}{! \Gamma_{\mathcal{B}}, \Pi \rightarrow r} (! \rightarrow)} (\text{contr, perm})^*$$

$\boxed{\Leftarrow}$  Recall that if  $\mathcal{B} = \{B_1, \dots, B_n\}$  is a finite set of formulae, then  $! \Gamma_{\mathcal{B}} = !B_1, \dots, !B_n$  (as stated above, the order of the elements in  $\mathcal{B}$  doesn't matter due to the  $(\text{perm}_{1,2})$  rules). For deriving sequents of the form  $! \Gamma_{\mathcal{B}}, \Pi \rightarrow C$ , where  $\Pi$ ,  $C$ , and  $\mathcal{B}$  do not contain  $!$  and  $\setminus$ , one can use a simpler calculus than  $! \mathbf{L}^*$ :

$$\frac{\frac{\overline{p \rightarrow p}}{! \Gamma_{\mathcal{B}}, \Pi \rightarrow A / B} (\rightarrow /)}{\frac{! \Gamma_{\mathcal{B}_1}, \Pi \rightarrow B \quad ! \Gamma_{\mathcal{B}_2}, \Delta_1, A, \Delta_2 \rightarrow C}{! \Gamma_{\mathcal{B}_1 \cup \mathcal{B}_2}, \Delta_1, A / B, \Pi, \Delta_2 \rightarrow C} (/ \rightarrow)} \quad \frac{! \Gamma_{\mathcal{B}}, \Delta_1, A, \Delta_2 \rightarrow C}{! \Gamma_{\mathcal{B} \cup \{A\}}, \Delta_1, \Delta_2 \rightarrow C} (! \rightarrow)$$

Moreover, the  $(! \rightarrow)$  rule is interchangeable with the others in the following ways:

$$\frac{\frac{! \Gamma_{\mathcal{B}}, \Delta_1, C, \Delta_2, B \rightarrow A}{! \Gamma_{\mathcal{B}}, \Delta_1, C, \Delta_2 \rightarrow A / B} (\rightarrow /)}{! \Gamma_{\mathcal{B} \cup \{C\}}, \Delta_1, \Delta_2 \rightarrow A / B} (! \rightarrow) \quad \rightsquigarrow \quad \frac{\frac{! \Gamma_{\mathcal{B}}, \Delta_1, C, \Delta_2, B \rightarrow A}{! \Gamma_{\mathcal{B} \cup \{C\}}, \Delta_1, \Delta_2, B \rightarrow A} (! \rightarrow)}{! \Gamma_{\mathcal{B} \cup \{C\}}, \Delta_1, \Delta_2 \rightarrow A / B} (\rightarrow /)$$

$$\frac{\frac{! \Gamma_{\mathcal{B}_1}, \Pi \rightarrow B \quad ! \Gamma_{\mathcal{B}_2}, \Delta_1, A, \Delta'_2, D, \Delta''_2 \rightarrow C}{! \Gamma_{\mathcal{B}_1 \cup \mathcal{B}_2}, \Delta_1, A / B, \Pi, \Delta'_2, D, \Delta''_2 \rightarrow C} (/ \rightarrow)}{! \Gamma_{\mathcal{B}_1 \cup \mathcal{B}_2 \cup \{D\}}, \Delta_1, A / B, \Pi, \Delta'_2, \Delta''_2 \rightarrow C} (! \rightarrow)$$

$\Downarrow$

$$\frac{! \Gamma_{\mathcal{B}_1}, \Pi \rightarrow B \quad \frac{! \Gamma_{\mathcal{B}_2}, \Delta_1, A, \Delta'_2, D, \Delta''_2 \rightarrow C}{! \Gamma_{\mathcal{B}_2 \cup \{D\}}, \Delta_1, A, \Delta'_2, \Delta''_2 \rightarrow C} (! \rightarrow)}{! \Gamma_{\mathcal{B}_1 \cup \mathcal{B}_2 \cup \{D\}}, \Delta_1, A / B, \Pi, \Delta'_2, \Delta''_2 \rightarrow C} (/ \rightarrow)$$

And the same, if  $D$  appears inside  $\Delta_1$  or  $\Pi$ . Finally, consecutive applications of  $(! \rightarrow)$  are always interchangeable.

After applying these transformations, we achieve a derivation where  $(! \rightarrow)$  is applied immediately after applying  $(/ \rightarrow)$  with the same active type (the other case, when it is applied after the axiom to  $p$ , is impossible, since  $\mathcal{B}$  is always a subset of  $\mathcal{G}_{\mathcal{R}}$ , and the latter doesn't contain sole variables). In other words, applications of  $(! \rightarrow)$  appear only in the following two situations:

$$\frac{\frac{! \Gamma_{\mathcal{B}_1}, \Pi \rightarrow p \quad ! \Gamma_{\mathcal{B}_2}, \Delta_1, r / q, \Delta_2 \rightarrow A}{! \Gamma_{\mathcal{B}_1 \cup \mathcal{B}_2}, \Delta_1, (r / q) / p, \Pi, \Delta_2 \rightarrow A} (/ \rightarrow)}{! \Gamma_{\mathcal{B}_1 \cup \mathcal{B}_2 \cup \{(r / q) / p\}}, \Delta_1, \Pi, \Delta_2 \rightarrow A} (! \rightarrow)$$

and

$$\frac{\frac{! \Gamma_{\mathcal{B}_1}, \Pi \rightarrow p/q \quad ! \Gamma_{\mathcal{B}_2}, \Delta_1, r, \Delta_2 \rightarrow A}{! \Gamma_{\mathcal{B}_1 \cup \mathcal{B}_2}, \Delta_1, r/(p/q), \Pi, \Delta_2 \rightarrow A} (/ \rightarrow)}{! \Gamma_{\mathcal{B}_1 \cup \mathcal{B}_2 \cup \{r/(p/q)\}}, \Delta_1, \Pi, \Delta_2 \rightarrow A} (! \rightarrow)$$

Now we prove the statement  $! \mathbf{L}^* \vdash ! \Gamma_{\mathcal{B}}, \Pi \rightarrow A$  (where  $\mathcal{B} \subseteq \mathcal{G}_{\mathcal{R}}\}) \Rightarrow \mathbf{L}^* + \mathcal{R} \vdash \Pi \rightarrow A$  by induction on the above canonical derivation. For the case of axiom or applications of rules  $(\rightarrow /)$  and  $(/ \rightarrow)$  we just apply the same rules in  $\mathbf{L}^* + \mathcal{R}$ , so the only interesting case is  $(! \rightarrow)$ . Consider the two possible situations.

In the  $(r/q)/p$  case, by induction hypothesis we get  $\mathbf{L}^* + \mathcal{R} \vdash \Pi \rightarrow p$  and  $\mathbf{L}^* + \mathcal{R} \vdash \Delta_1, r/q, \Delta_2 \rightarrow A$ , and then we develop the following derivation in  $\mathbf{L}^* + \mathcal{R}$  (recall that (cut) is admissible there):

$$\frac{\Pi \rightarrow p \quad \frac{\frac{p, q \rightarrow r}{p \rightarrow r/q} (\rightarrow /) \quad \Delta_1, r/q, \Delta_2 \rightarrow A}{\Delta_1, p, \Delta_2 \rightarrow A} (\text{cut})}{\Delta_1, \Pi, \Delta_2 \rightarrow A} (\text{cut})$$

In the case of  $r/(p/q)$ , the derivation looks like this:

$$\frac{\Pi \rightarrow p/q \quad \frac{p/q \rightarrow r \quad \Delta_1, r, \Delta_2 \rightarrow A}{\Delta_1, p/q, \Delta_2 \rightarrow A} (\text{cut})}{\Delta_1, \Pi, \Delta_2 \rightarrow A} (\text{cut})$$

This completes the proof of Theorem 5.  $\square$

Now we can return to our main claim.

*Proof of Theorem 1.* Take  $\mathcal{R}_0$  from Theorem 4 and suppose that  $! \mathbf{L}^*$  is decidable. Then we can present an algorithm that solves the derivability problem for  $\mathbf{L}^* + \mathcal{R}_0$ . Namely, for a sequent  $\Pi \rightarrow A$  we search through all subsets  $\mathcal{B} \subseteq \mathcal{G}_{\mathcal{R}}$  (and there is a finite number of them) and test derivability of  $! \Gamma_{\mathcal{B}}, \Pi \rightarrow A$  in  $! \mathbf{L}^*$ . By Theorem 5,  $\Pi \rightarrow A$  is derivable in  $\mathbf{L}^* + \mathcal{R}_0$  if and only if at least one of these tests succeeds. This contradicts Theorem 4. Therefore  $! \mathbf{L}^*$  is undecidable.

Since we never used  $\setminus$  in the construction, we get undecidability for  $! \mathbf{L}^*_\setminus$ .  $\square$

This proof of Theorem 1 is in the spirit of our previous work [7]. The significant difference between this paper and [7] is that here the modality does not satisfy the weakening rule and the system  $! \mathbf{L}^*$  doesn't obey any version of Lambek's restriction (*i.e.*, the antecedents are allowed to be empty). Due to the lack of the weakening rule, in Theorem 5 it is not sufficient to check derivability only for  $\mathcal{B} = \mathcal{G}_{\mathcal{R}}$ , and therefore Theorem 5 is formulated in the relevant logic style. We also had to open up and reassemble Buszkowski's proof from [2] and [3] to make it work without Lambek's restriction (in  $\mathbf{L}^*$ ).

## 5 A Decidable Fragment of $!L^*$

Undecidability of  $!L^*$  is somewhat unfortunate, because this calculus is linguistically motivated (see Section 2). However, in our examples  $!$  was applied only to primitive types ( $np$ ). If we consider only sequents with this restriction, the situation is different: the derivability problem becomes decidable. Moreover, it belongs to NP.

Let's call the *size* of a formula  $A$  (denoted by  $|A|$ ) the total number of variable and connective occurrences in  $A$ . More formally,  $|A|$  is defined recursively:  $|p| = 1$  for  $p \in \text{Var}$ ,  $|A \setminus B| = |B / A| = |A| + |B| + 1$ ,  $!A = |A| + 1$ . The size of a sequent  $A_1, \dots, A_n \rightarrow B$  is  $|A_1| + \dots + |A_n| + |B|$ .

In the pure Lambek calculus, the size of any derivation is necessarily bounded by the size of the goal sequent. In our case, a sequent could have derivations of arbitrary size due to uncontrolled application of permutation rules: two consecutive applications of  $(\text{perm}_1)$  and  $(\text{perm}_2)$  (with the same formula at the same places) do nothing with the sequent, but increase the derivation size. Nevertheless, the following lemma shows that every sequent has a derivation of quadratic size.

**Lemma 1.** *If the sequent  $\Pi \rightarrow C$  is derivable in  $!L^*$  and  $!$  in this sequent is applied only to variables, then this sequent has a derivation of size less than  $12n^2 + 3n$ , where  $n$  is the size of  $\Pi \rightarrow C$ .*

Recall that (cut) is not included in the system, all derivations are cut-free.

*Proof.* We represent the derivation of  $\Pi \rightarrow C$  as a tree. The leaves of the tree are instances of axioms, and the inner nodes correspond to applications of rules. Rules  $(/ \rightarrow)$  and  $(\setminus \rightarrow)$  form *branching points* of the tree. The number of leaves is equal to the number of branching points plus one.

Let's call  $(\text{perm}_{1,2})$  and  $(\text{contr})$  *structural* rules; other rules are *logical* ones.

Each logical rule introduces exactly one connective into the goal sequent  $\Pi \rightarrow C$ . The key note here is the fact that, since only variables can appear under  $!$ , the contraction rule  $(\text{contr})$  cannot merge two connectives. Therefore, since the total number of connectives is less than  $n$ , the number of logical rule applications is also less than  $n$ .

Each branching point corresponds to an application of a logical rule, whence the number of branching points is also less than  $n$ . Therefore, in the tree there are no more than  $n$  axiom leaves, and each axiom introduces two variable occurrences. Let's trace these occurrences down the tree. Each occurrence either traces to an occurrence in the goal sequent, or disappears (gets merged with another occurrence) in an application of  $(\text{contr})$ . Thus, the number of  $(\text{contr})$  applications is less than the total number of variable occurrences in axiom leaves, and, therefore, less than  $2n$ .

Finally, we limit the number of  $(\text{perm}_{1,2})$  applications. As said above, a block of consecutive applications of  $(\text{perm}_{1,2})$  can include an arbitrarily large number of  $(\text{perm}_{1,2})$  applications. However, we can always reduce it. Each

block of consecutive permutations has the following form:

$$\frac{\Delta_1, !A_1, \Delta_2, !A_2, \Delta_3, \dots, \Delta_k, !A_k, \Delta_{k+1} \rightarrow B}{\Delta'_1, !A_{i_1}, \Delta'_2, !A_{i_2}, \Delta'_3, \dots, \Delta'_k, !A_{i_k}, \Delta'_{k+1} \rightarrow B} (\text{perm}_{1,2})^*,$$

where the sequences  $\Delta_1, \dots, \Delta_{k+1}$  and  $\Delta'_1, \dots, \Delta'_{k+1}$  coincide and  $\{i_1, \dots, i_k\} = \{1, \dots, k\}$ .

The number of formulae in the left-hand side of the sequent here is bounded by  $3n$  (it was less than  $n$  in the goal sequent  $\Pi \rightarrow C$ , and, in the worst case, it was increased by less than  $2n$  applications of (contr)). Therefore,  $k < 3n$ . Now we replace this block with a block of  $k$  permutations: each  $!A_i$  is moved to its place by one permutation. Thus, in each block we have less than  $3n$  permutations.

Each  $(\text{perm}_{1,2})$  block is preceded by an application of a rule different from  $(\text{perm}_{1,2})$  or an axiom leaf. Thus the number of such blocks is bounded by  $4n$  ( $n$  for logical rules,  $2n$  for contractions,  $n$  for axioms).

Therefore, the number of  $(\text{perm}_{1,2})$  applications is less than  $12n^2$ , and the total size of the derivation is less than  $12n^2 + 3n$ .  $\square$

This lemma yields the following decidability result:

**Theorem 6.** *The derivability problem in  $!\mathbf{L}^*$  for sequents in which  $!$  is applied only to variables is decidable and belongs to the NP class (i.e., can be resolved by a nondeterministic polynomial algorithm).*

## 6 Future Work

Since the Lambek calculus itself is NP-complete [14][17], we get NP-completeness of  $!\mathbf{L}^*$  in the restricted case, where  $!$  can be applied only to variables. On the other hand, it is known that the derivability problem for the fragment of the pure Lambek calculus with only one division operation is decidable in polynomial time [16][15]. Therefore, the complexity for the restricted case of  $!\mathbf{L}^*_/\backslash$  (where we have only one division, and  $!$  can be applied only to variables) yet should be studied. It belongs to NP (by our Theorem 6), and the question is whether this fragment is poly-time decidable or NP-hard. Recall that in the unrestricted case we've proved undecidability not only for the whole  $!\mathbf{L}^*$ , but also for its one-division fragment,  $!\mathbf{L}^*_/\backslash$ .

Another interesting question is whether our decidability result (Theorem 6) can be extended to the situation where  $!$  can be applied to formulae of Horn depth 1, i.e., formulae, in which all denominators of  $/$  and  $\backslash$  are primitive types, for instance,  $(p \backslash (q / r)) / s$ . Notice that if we allow formulae of Horn depth 2 (of the form  $r / (p / q)$ ) under  $!$ , then we immediately get undecidability (see Section 4).

Our encoding in Theorem 5 actually shows that *grammars* based on  $!\mathbf{L}^*$  can generate arbitrary recursively enumerable languages. On the other hand, pure Lambek grammars generate precisely context-free languages [13]. Moreover,

this holds also in the so-called strong sense, *i.e.*, context-free grammars and Lambek grammars can assign the same Montague-style semantic values to the words derived [6][8]. The question is what class of grammars in the Chomsky hierarchy corresponds to grammars based on the fragment of  $\mathbf{!L}^*$ , restricted as in Theorem 6, and could one add Montague-style semantics to such grammars.

## Acknowledgements

Stepan Kuznetsov’s research was supported by the Russian Foundation for Basic Research (grants 15-01-09218-a and 14-01-00127-a) and by the Presidential Council for Support of Leading Scientific Schools (grant NŠ-9091.2016.1). Max Kanovich’s research was partially supported by EPSRC. Andre Scedrov’s research was partially supported by ONR.

This research was performed in part during visits of Stepan Kuznetsov and Max Kanovich to the University of Pennsylvania. We greatly appreciate support of the Mathematics Department of the University. A part of the work was also done during the stay of Andre Scedrov at the National Research University Higher School of Economics. We would like to thank S. O. Kuznetsov and I. A. Makarov for hosting there.

The paper was prepared in part within the framework of the Basic Research Program at the National Research University Higher School of Economics (HSE) and was partially supported within the framework of a subsidy by the Russian Academic Excellence Project ‘5–100’.

We are indepted to the participants of the research seminars “Logical Problems in Computer Science” and “Algorithmic Problems in Algebra and Logic” at Moscow (Lomonosov) University, in particular, S. I. Adian, L. D. Beklemishev, V. N. Krupski, I. I. Osipov, F. N. Pakhomov, M. R. Pentus, D. S. Shamkanov, I. B. Shapirovsky, V. B. Shehtman, A. A. Sorokin, T. L. Yavorskaya, and others for fruitful discussions and suggestions that allowed us to improve our presentation significantly.

## References

- [1] V. M. Abrusci. A comparison between Lambek syntactic calculus and intuitionistic linear propositional logic. *Zeitschr. für math. Logik und Grundl. der Math. (Math. Logic Quart.)*, Vol. 36, 1990. P. 11–15.
- [2] W. Buszkowski. Some decision problems in the theory of syntactic categories. *Zeitschr. für math. Logik und Grundl. der Math. (Math. Logic Quart.)* Vol. 28, 1982. P. 539–548.
- [3] W. Buszkowski. Lambek calculus with nonlogical axioms. *Language and Grammar (CSLI Lect. Notes, vol. 168)*, 2005. P. 77–93.
- [4] B. Carpenter. *Type-logical semantics*. MIT Press, 1998.



- [5] J.-Y. Girard. Linear logic. *Theor. Comput. Sci.*, Vol. 50, No. 1, 1987. P. 1–102.
- [6] M. Kanazawa, S. Salvati. The string-meaning relations definable by Lambek grammars and context-free grammars. *Proc. Formal Grammar '12/'13* (LNCS vol. 8036), Springer, 2013. P. 191–208.
- [7] M. Kanovich, S. Kuznetsov, A. Scedrov. On Lambek's restriction in the presence of exponential modalities. *Proc. LFCS '16* (LNCS vol. 9537), Springer, 2015. P. 146–158.
- [8] S. L. Kuznetsov. On translating context-free grammars into Lambek grammars. *Proc. Steklov Inst. Math.*, Vol. 290, 2015. P. 63–69.
- [9] J. Lambek. The mathematics of sentence structure. *Amer. Math. Monthly*, Vol. 65, No. 3, 1958. P. 154–170.
- [10] J. Lambek. On the calculus of syntactic types. *Structure of Language and Its Mathematical Aspects* (Proc. Symposia Appl. Math., vol. 12), AMS, 1961. P. 166—178.
- [11] P. Lincoln, J. Mitchell, A. Scedrov, N. Shankar. Decision problems for propositional linear logic. *Annals of Pure and Applied Logic*, Vol. 56, Iss. 1–3, 1992. P. 239–311.
- [12] G. Morrill, O. Valentín. Computational coverage of TLG: Nonlinearity. *Proc. NLCS '15* (EPiC Series, vol. 32), 2015. P. 51–63
- [13] M. Pentus. Product-free Lambek calculus and context-free grammars. *J. Symb. Log.*, Vol. 62, No. 2, 1997. P. 648–660.
- [14] M. Pentus. Lambek calculus is NP-complete. *Theor. Comput. Sci.*, Vol. 357, 2006. P. 186–201.
- [15] M. Pentus. Complexity of the Lambek calculus and its fragments. *Advances in Modal Logic*, vol. 8, College Publications, 2010. P. 310–329.
- [16] Yu. Savateev. Lambek grammars with one division are decidable in polynomial time. *Proc. CSR '08* (LNCS vol. 5010), Springer, 2008. P. 273–282.
- [17] Yu. Savateev. Product-free Lambek calculus is NP-complete. *Proc. LFCS '09* (LNCS vol. 5407), Springer, 2009. P. 380–394.
- [18] D. N. Yetter. Quantales and (noncommutative) linear logic. *J. Symb. Log.* Vol. 55, No. 1, 1990. P. 41–64.